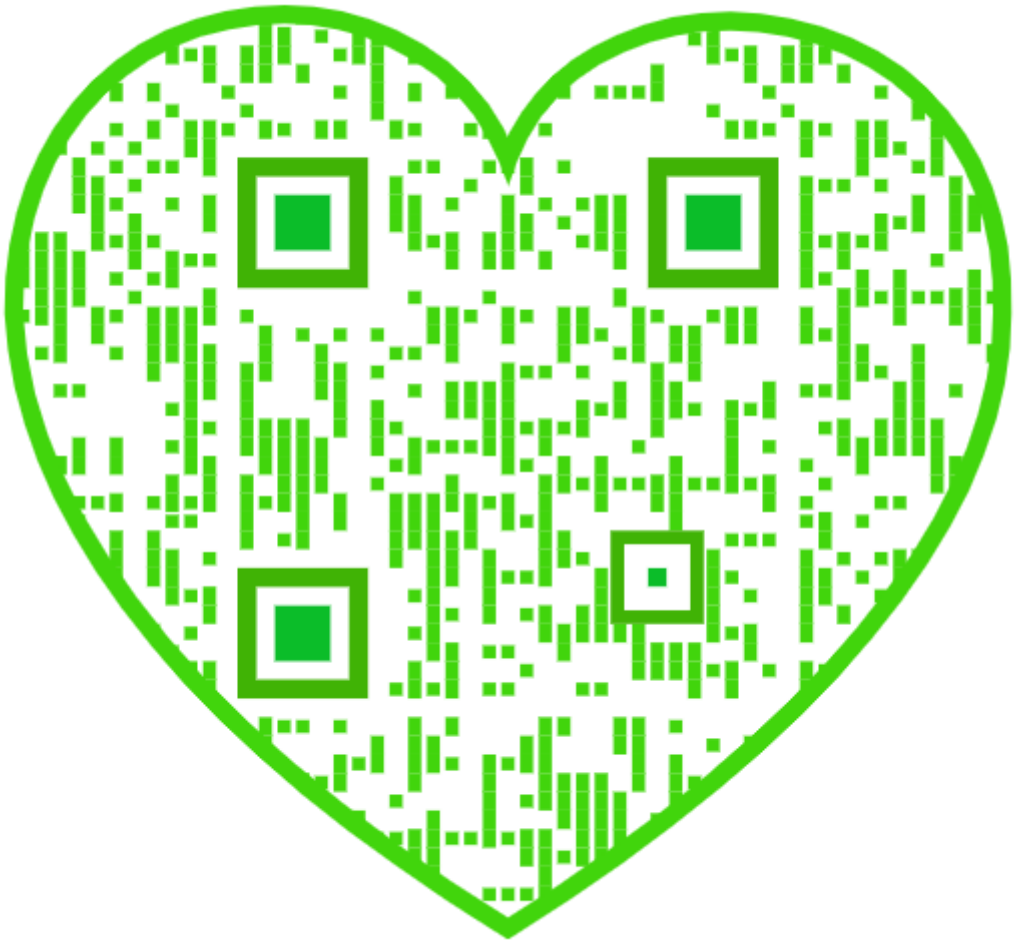
Master in Artificial Intelligence

Data Collection & Preprocessing V



Purpose

The purpose of the section is to help you learn how to collect and preprocess data to become a Successful Artificial Intelligence (AI) Engineer

At the end of this lecture, you will learn the following

- **An example of gathering relevant data from various sources, ensure its quality, and preprocess it to make it suitable for analysis and modeling**

How to collect and preprocess data- An Example

Gather relevant data

Ensure its quality

Preprocess it

Make it suitable for analysis and modeling.

Gathered Relevant Data

Online review platforms

Social media platforms

Customer feedback surveys

Public datasets

Ensured Data Quality

- Data integrity
- Data accuracy
- Language and sentiment relevance
- Spam detection

Preprocessed Data

- Text cleaning
- Tokenization
- Stopword removal
- Lemmatization or stemming
- Handling emojis and emoticons
- Sentiment labeling

Feature Extraction

Bag-of-Words (BoW) representation

TF-IDF (Term Frequency-Inverse Document Frequency)

Word embeddings

What was done in BOW representation

Bag-of-Words (BoW) representation

TF-IDF (Term Frequency-Inverse Document Frequency)

Word embeddings

What was done in BOW representation

Tokenization

↓

Vocabulary Creation

↓

Counting Word Frequencies

↓

Constructing the Feature Matrix

What was done in BOW representation-Example

- We had three preprocessed reviews as an example:
  - "The product is great and works well."
  - "I am satisfied with my purchase."
  - "This product is terrible and does not work."
- The vocabulary created from these reviews included words like "product", "great", "works", "well", "satisfied", "purchase", "terrible", "does", "not", etc.
- Using this vocabulary, we constructed the BoW feature matrix

Document	product	great	works	well	satisfied	purchase	terrible	does	not
Review 1	1	1	1	1	0	0	0	0	0
Review 2	1	0	0	0	1	1	0	0	0
Review 3	1	0	1	0	0	0	1	1	1

What did TF-IDF scores look like?

Bag-of-Words (BoW) representation

TF-IDF (Term Frequency-Inverse Document Frequency)

Word embeddings

What did TF-IDF scores look like?

Let's continue with the example of three preprocessed reviews:
1."The product is great and works well."
2."I am satisfied with my purchase."
3."This product is terrible and does not work.“

Calculated the TF-IDF scores for each word in the vocabulary. The TF-IDF score for a word in a document was computed as the product of its term frequency (TF) and inverse document frequency (IDF).

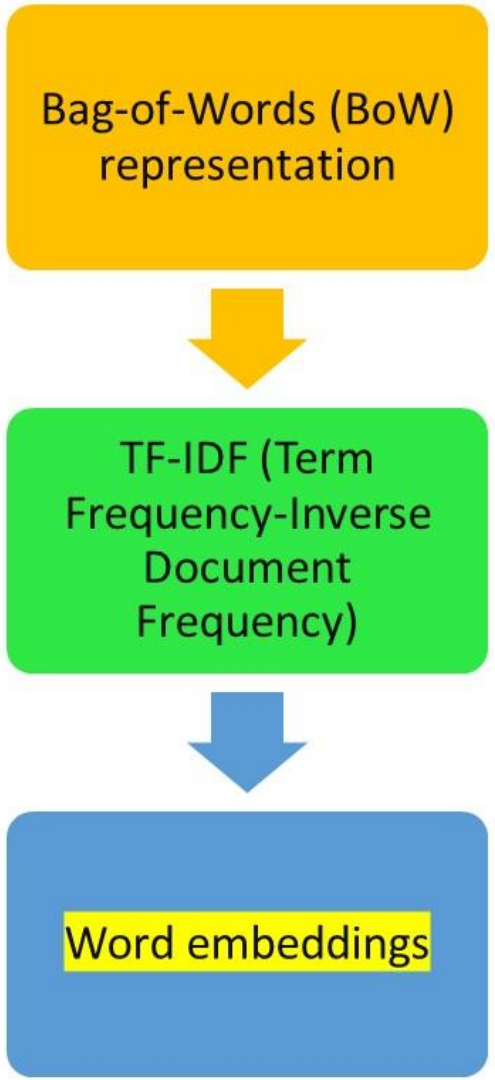
1.Term Frequency (TF):
    1. The term frequency of a word in a document is the number of times the word appears in the document divided by the total number of words in the document.
2.Inverse Document Frequency (IDF):
    1. The inverse document frequency of a word measures how unique or rare the word is across all documents in the corpus. It is calculated as the logarithm of the ratio of the total number of documents to the number of documents containing the word

What did TF-IDF scores look like?

Term	Review 1 (TF-IDF)	Review 2 (TF-IDF)	Review 3 (TF-IDF)
product	tfidf(product, Review 1) = 1 * idf(product) = 1 * log(3/3) = 0	tfidf(product, Review 2) = 1 * log(3/3) = 0	tfidf(product, Review 3) = 1 * log(3/3) = 0
great	tfidf(great, Review 1) = 1 * log(3/1) = log(3)	tfidf(great, Review 2) = 0	tfidf(great, Review 3) = 0
works	tfidf(works, Review 1) = 1 * log(3/1) = log(3)	tfidf(works, Review 2) = 0	tfidf(works, Review 3) = 1 * log(3/1) = log(3)
well	tfidf(well, Review 1) = 1 * log(3/1) = log(3)	tfidf(well, Review 2) = 0	tfidf(well, Review 3) = 0
satisfied	tfidf(satisfied, Review 1) = 0	tfidf(satisfied, Review 2) = 1 * log(3/1) = log(3)	tfidf(satisfied, Review 3) = 0
purchase	tfidf(purchase, Review 1) = 0	tfidf(purchase, Review 2) = 1 * log(3/1) = log(3)	tfidf(purchase, Review 3) = 0
terrible	tfidf(terrible, Review 1) = 0	tfidf(terrible, Review 2) = 0	tfidf(terrible, Review 3) = 1 * log(3/1) = log(3)
does	tfidf(does, Review 1) = 0	tfidf(does, Review 2) = 0	tfidf(does, Review 3) = 1 * log(3/1) = log(3)
not	tfidf(not, Review 1) = 0	tfidf(not, Review 2) = 0	tfidf(not, Review 3) = 1 * log(3/1) = log(3)

What is next?

How pre-trained word embeddings like Word2Vec or GloVe used to represent words as dense vectors in a continuous vector space



Master in Artificial Intelligence

Data Collection & Preprocessing V